A Method for Compensation of Interactions Between Second-Order Actuators and Control Allocators

Michael W. Oppenheimer, Member
David B. Doman, Member
Control Design and Analysis Branch
2210 Eighth St., Bldg. 146, Rm. 305
Wright-Patterson AFB, OH, USA
937-255-8490
michael.oppenheimer@wpafb.af.mil

Abstract—Typically, actuator dynamics are ignored when designing flight control allocators for aircraft because the bandwidths of actuators are normally much larger than the frequencies of the vehicle's rigid body modes. Unfortunately, this in not always the case, particularly when dealing with non-aerodynamic surface actuators. Ignoring the interactions between constrained control allocators and actuator dynamics can have serious consequences. In this work, a method, which post-processes the output of a control allocation algorithm, is developed to compensate for actuator dynamics. The actuators can have dynamics which are either first-order, second-order with no zeros, or second-order with a single zero. The method developed here solves for a gain, which multiplies the commanded change in control effector setting as computed by the control allocator. This approach is not computationally intensive and thus has the added benefit of being an algorithm which can operate in real-time on a typical flight computer. Likewise, this approach is applicable to both the saturated and unsaturated control effector cases. The basic premise of this method is to post-process the output of the control allocation algorithm to overdrive the actuators so that at the end of a sampling interval, the actual actuator positions are equivalent to the desired actuator positions.

TABLE OF CONTENTS

1. INTRODUCTION

Control allocation algorithms are used to compute aircraft control surface deflections which produce a desired set of moment or acceleration commands. A number of approaches have been developed that ensure that the commands provided to the effectors are physically realizable. These actuator command signals are feasible in the sense that they do not exceed hardware rate and position limits. Buffington[1] and Bodson[2] developed methods which take into account rate and position limits.

A review of the constrained control allocation literature shows that the coupling effects that result from combining constrained control allocators and actuator dynamics has largely been ignored, although some recent research has been performed on directly including actuator dynamics in the control allocation problem[3], [4]. The underlying assumption of most previous work is that actuators respond instantaneously to commands. This assumption may at first seem justified because in practice, actuator dynamics are typically much faster than the rigid body modes that are to be controlled. However, interactions between a constrained control allocator and an actuator with linear dynamics can result in a system that falls well short of its potential. As an example, a simulation was run with a linear programming based control allocation algorithm mixing four control effectors to obtain a desired set of moments, $\mathbf{d}_{des} \in \mathbb{R}^3$. In this simulation run, the actuator dynamics for each control effector were set to $\frac{\delta(s)}{\delta_{cmd}(s)} = \frac{5}{s+5}$ and the commanded effector positions, as computed by the control allocator, were used to initialize the allocator at the next timestep. To illustrate the effects of constrained control allocator and actuator dynamics interactions, consider Figures 1 and 2. The objective is to make the accelerations produced by the control effectors ($B\boldsymbol{\delta}$) equal to the commanded accelerations ($\mathbf{d}_{des}$). Figure 1 shows that when there are no actuator dynamics, the desired result is achieved, namely $B\boldsymbol{\delta} = \mathbf{d}_{des}$. When actuator dynamics are included, the results are as shown in Figure 2 where it is obvious that $B\boldsymbol{\delta} \neq \mathbf{d}_{des}$. The goal of this work is to develop a scheme, which compensates for the effects of actuator dynamics, so that even when actuator dynamics are included, results like those shown in Figure 1 are still obtained. As was shown in the example above, the control allocator/actuator interac-

0-7803-8870-4/05/$20.00/©2005 IEEE.
IEEEAC paper # 1164

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **2005**	2. REPORT TYPE	3. DATES COVERED **00-00-2005 to 00-00-2005**

4. TITLE AND SUBTITLE **A Method for Compensation of Interactions Between Second-Order Actuators and Control Allocators**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Air Force Research Laboratory,Air Vehicles Directorate,Wright Patterson AFB,OH,45433**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES **8**	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**			

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

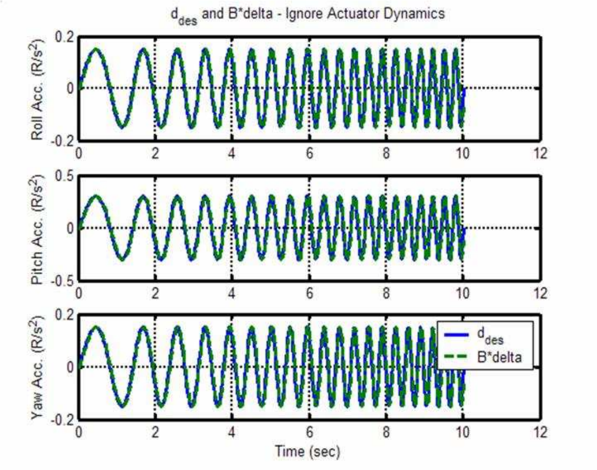

Figure 1. Acceleration Commands and Accelerations Produced by the Controls - No Actuator Dynamics.

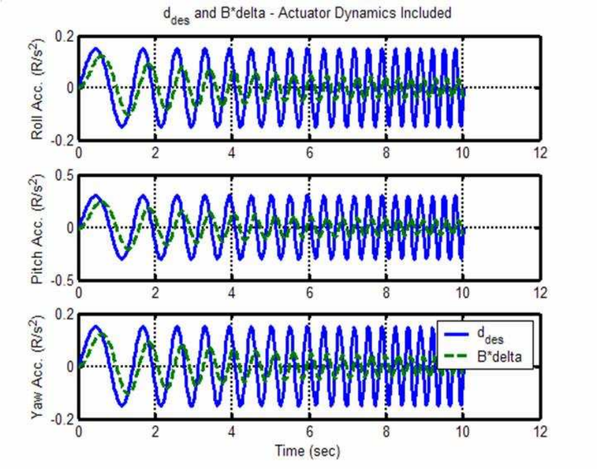

Figure 2. Acceleration Commands and Accelerations Produced by the Controls - Actuator Dynamics Present.

tion can yield control effector positions which produce significantly different accelerations than the commanded accelerations. Bolling[5] has shown that the interaction between first-order actuator dynamics and constrained control allocation algorithms can be eliminated by overdriving the actuators. In previous work, [6] the effects of the control allocator/actuator interaction for first-order and second-order (with no zeros) actuator dynamics has been taken into account. That work is expanded here to include second-order actuator dynamics with a single zero. Having the first-order, second-order, and second-order with a zero cases defined, this work makes it possible to create higher-order actuators as a combination of these simpler forms.

In this work, results for the first-order and simple second-order cases are provided to illustrate the effects of actuator dynamics on control system. Following this, the details of the interaction between constrained control allocators and second-order actuator dynamics with a zero are provided in detail. Simulation results are presented for a four control surface vehicle using a linear programming based control allocation algorithm that takes into account control effector position and rate limits. The results show that the desired control effector positions can be obtained by post-processing the control allocation commands.

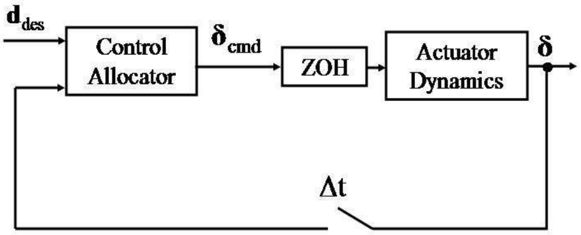

Figure 3. Control allocator and actuator interconnection.

2. System Definition

Figure 3 shows the system to be analyzed in this work. Inputs to the control allocation algorithm consist of a vector of desired moment or acceleration commands, $\mathbf{d}_{des} \in \mathbb{R}^n$, and a vector containing the current control surface deflections, $\boldsymbol{\delta} \in \mathbb{R}^m$. The output of the control allocator is the commanded control surface deflection vector, $\boldsymbol{\delta}_{cmd} \in \mathbb{R}^m$. The actuator dynamics respond to $\boldsymbol{\delta}_{cmd}$ to produce the actual control deflections, $\boldsymbol{\delta}$. The individual actuators are assumed to have uncoupled dynamics, hardware rate limits $\pm\dot{\boldsymbol{\delta}}_{\max}$, and position limits $\boldsymbol{\delta}_{\min}, \boldsymbol{\delta}_{\max}$. In most control allocator implementations, rate limits are taken into account by converting them into effective position limits at the end of the next sampling period and constraining the effector commands to respect the most restrictive of the rate or position limits, i.e.,

$$\begin{aligned} \overline{\boldsymbol{\delta}} &= \min(\boldsymbol{\delta}_{\max}, \boldsymbol{\delta} + \dot{\boldsymbol{\delta}}_{\max}\Delta t) \\ \underline{\boldsymbol{\delta}} &= \max(\boldsymbol{\delta}_{\min}, \boldsymbol{\delta} - \dot{\boldsymbol{\delta}}_{\max}\Delta t) \end{aligned} \tag{1}$$

where $\boldsymbol{\delta}$ is the current location of the control effectors, $\overline{\boldsymbol{\delta}}, \underline{\boldsymbol{\delta}}$ are the most restrictive upper and lower bounds on the effectors, respectively, and Δt is the sampling period of the digital flight control system.

A few comments are in order regarding Figure 3 and the analysis described above. First, note that the instantaneous position limit given by Equation 1 makes use of a sampled vector of actuator position measurements to compute the maximum distance that the actuator can move in the next time instant. This is in contrast to using the previous value of actuator command vector, $\boldsymbol{\delta}_{cmd}$, as is often done in simulation. The motivation for using actuator measurements is that when actuator dynamics, disturbances, and uncertainties are taken into account, the actuator command vector and the true actuator positions will differ. This difference can cause a control allocator to generate inappropriate actuator commands that do not deliver the desired moments or accelerations. Thus, feeding the measured actuator position vector to the control allocator has the advantage of reducing uncertainty in the actuator position. It is important; however, to be aware of the effects of using the measured actuator position vector.

3. Attenuation of Zero-Order-Hold Inputs For First-Order and Second-Order (No Zeros) Actuator Dynamics

Referring to Figure 3, the desired situation would be for $\boldsymbol{\delta} = \boldsymbol{\delta}_{cmd}$. However, actuator dynamics alter the command signals so that, in general, $\boldsymbol{\delta} \neq \boldsymbol{\delta}_{cmd}$. For actuators with high bandwidths relative to the rigid body modes, this is not a serious concern. However, situations exist where the actuator dynamics are not sufficiently fast and need to be taken into account. In this section, the effects of first-order actuator dynamics and simple second-order (no zero) actuator dynamics on the system shown in Figure 3 will be discussed. Let the dynamics of a single actuator be represented by a continuous time first-order transfer function of the form

$$\frac{\delta(s)}{\delta_{cmd}(s)} = \frac{a}{s+a} \tag{2}$$

The discrete time representation of the first-order actuator dynamics equation is given by

$$\delta(t_{k+1}) = \Phi\delta(t_k) + \Gamma\delta_{cmd}(t_k) \tag{3}$$

where $\Phi \triangleq e^{-a\Delta t}$, $\Gamma \triangleq 1 - e^{-a\Delta t}$, and it has been assumed that the input to the actuator dynamics, $\delta_{cmd}(t_k)$, is held constant over each sampling period. The command to the actuator can be written as

$$\delta_{cmd}(t_k) = \Delta\delta_{cmd_{CA}}(t_k) + \delta(t_k) \tag{4}$$

where the commanded incremental change in actuator position over one timestep is defined by $\Delta\delta_{cmd_{CA}}(t_k) \triangleq \delta_{cmd_{CA}}(t_k) - \delta(t_k)$ and where $\delta_{cmd_{CA}}(t_k)$ is the actuator position command from the control allocator. Since the effector commands are held constant over each sampling period, $\Delta\delta_{cmd_{CA}}(t_k)$ will appear to the actuators to be a step command from the measured position. Recall that $\delta_{cmd_{CA}}(t_k)$ is calculated by the control allocator based upon the assumption that the effector will respond instantaneously to commands. Substituting Equation 4 into Equation 3 yields

$$\delta(t_{k+1}) = \Phi\delta(t_k) + \Gamma\left[\Delta\delta_{cmd_{CA}}(t_k) + \delta(t_k)\right] \tag{5}$$

Since $\Gamma < 1$, the incremental command signal from the control allocation algorithm, $\Delta\delta_{cmd_{CA}}(t_k)$ is attenuated by the actuator dynamics, thus $\delta(t_{k+1}) \neq \delta_{cmd_{CA}}(t_k)$. The objective is to find a gain, M, that modifies the output of the control allocation algorithm such that $\delta(t_{k+1}) = \delta_{cmd_{CA}}(t_k) = \Delta\delta_{cmd_{CA}}(t_k) + \delta(t_k)$. Hence,

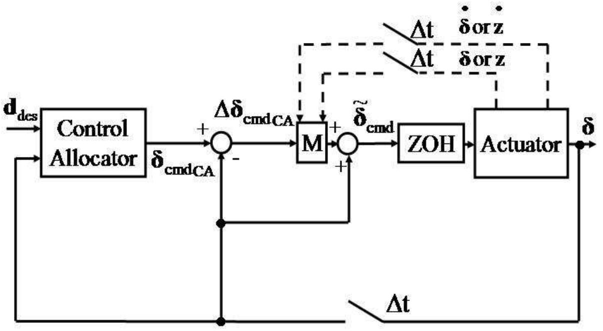
$\dot{\delta}$ or $\dot{z}$
Δt
δ or z
Δt
$\mathbf{d}_{\mathbf{dcs}}$
Control Allocator
$\boldsymbol{\delta}_{\mathbf{cmdCA}}$
$\Delta\boldsymbol{\delta}_{\mathbf{cmdCA}}$
+
-
M
+
+
$\tilde{\boldsymbol{\delta}}_{\mathbf{cmd}}$
ZOH
Actuator
$\boldsymbol{\delta}$
Δt

Figure 4. Block diagram of command increment compensation.

$$\delta(t_{k+1}) = \Phi\delta(t_k) + \Gamma\left[M\Delta\delta_{cmd_{CA}}(t_k) + \delta(t_k)\right] \tag{6}$$

and, solving for M yields

$$M = \frac{1}{\Gamma} \tag{7}$$

Thus the actuator command signal must be modified such that

$$\begin{aligned}\tilde{\delta}_{cmd}(t_k) &= M\Delta\delta_{cmd_{CA}}(t_k) + \delta(t_k)\\ &= \frac{1}{\Gamma}\Delta\delta_{cmd_{CA}}(t_k) + \delta(t_k)\end{aligned} \tag{8}$$

Replacing $\delta_{cmd}(t_k)$ in Equation 3 with $\tilde{\delta}_{cmd}(t_k)$ from Equation 8 yields

$$\begin{aligned}\delta(t_{k+1}) &= \Phi\delta(t_k) + \Gamma\tilde{\delta}_{cmd}(t_k)\\ &= \Phi\delta(t_k) + \Gamma\left[\frac{1}{\Gamma}\Delta\delta_{cmd_{CA}}(t_k) + \delta(t_k)\right]\end{aligned} \tag{9}$$

which yields the desired actuator position.

Since Γ can be computed from the known quantities a and Δt, one can compensate for command increment attenuation using Equation 8. For a bank of decoupled first-order actuators with nominal bandwidths of a_i, corresponding values of Γ_i can be computed using $\Gamma_i = (1 - e^{-a_i\Delta t})$. The command increment compensation can then be implemented in discrete time as shown in Figure 4. In Figure 4, the dashed lines are for compensation for second-order actuator dynamics, as will be discussed shortly, and therefore, are not part of the first-order actuator dynamics compensation scheme. Note that, for multiple actuators, **M** in Figure 4 is a diagonal matrix with the entries on the main diagonal being $\Gamma_1, \Gamma_2, \cdots, \Gamma_m$, where the subscript m is defined as the number of control effectors. Hence, the magnitude of the control allocation command increment is modified to counteract the attenuation that results from the interaction between first-order actuator dynamics and the control allocator.

With the knowledge gained from the first-order actuator dynamics case, it is now possible to extend the previous results to second-order actuator dynamics. Let the second-order actuator dynamics be represented by the general equation

$$\frac{\delta(s)}{\delta_{cmd}(s)} = \frac{k}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{10}$$

which, in state-space form, becomes

$$\begin{aligned} \begin{bmatrix} \dot{\delta}(t) \\ \ddot{\delta}(t) \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \begin{bmatrix} \delta(t) \\ \dot{\delta}(t) \end{bmatrix} \\ &+ \begin{bmatrix} 0 \\ k \end{bmatrix} \delta_{cmd}(t) \\ &= \mathbf{A} \begin{bmatrix} \delta(t) \\ \dot{\delta}(t) \end{bmatrix} + \mathbf{B}\delta_{cmd}(t) \end{aligned} \tag{11}$$

$$\delta(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \delta(t) \\ \dot{\delta}(t) \end{bmatrix} = \mathbf{C} \begin{bmatrix} \delta(t) \\ \dot{\delta}(t) \end{bmatrix} \tag{12}$$

Using Equations 11 and 12, the discrete-time solution to the second-order actuator dynamics differential equation becomes

$$\begin{aligned} \delta(t_{k+1}) &= \begin{bmatrix} \Phi_{1,1} & \Phi_{1,2} \end{bmatrix} \begin{bmatrix} \delta(t_k) \\ \dot{\delta}(t_k) \end{bmatrix} \\ &+ \delta_{cmd}(t_k) \int_{t_k}^{t_{k+1}} k\Phi_{1,2}(t_{k+1} - \tau) d\tau \end{aligned} \tag{13}$$

where

$$\mathbf{\Phi} = e^{\mathbf{A}(t_{k+1} - t_k)} = \begin{bmatrix} \Phi_{1,1} & \Phi_{1,2} \\ \Phi_{2,1} & \Phi_{2,2} \end{bmatrix} \tag{14}$$

is the state transition matrix. Since **A** is constant, the state transition matrix depends on the time difference, $\Delta t = t_{k+1} - t_k$, and the explicit dependence on time has been eliminated. Performing the operations required in Equations 13 and 14 produces

$$\delta(t_{k+1}) = C_1\delta(t_k) + C_2\dot{\delta}(t_k) + C_3\delta_{cmd}(t_k) \tag{15}$$

where

$$\begin{aligned} C_1 &= \frac{\omega_n}{\omega_d} e^{\sigma\Delta t} sin \left[\omega_d \Delta t + \arctan\left(\frac{\omega_d}{-\sigma} \right) \right] \\ C_2 &= \frac{e^{\sigma\Delta t}}{\omega_d} sin \left[\omega_d \Delta t \right] \\ C_3 &= \frac{k}{\omega_d} \left[\frac{\omega_d + e^{\sigma\Delta t} \left[\sigma sin(\omega_d \Delta t) - \omega_d cos(\omega_d \Delta t) \right]}{\omega_n^2} \right] \end{aligned} \tag{16}$$

$\omega_d = \omega_n \sqrt{1 - \zeta^2}$ and $\sigma = -\zeta\omega_n$. Substituting Equation 4 into Equation 15 produces

$$\delta(t_{k+1}) = C_1\delta(t_k) + C_2\dot{\delta}(t_k) + C_3 \left[\Delta\delta_{cmd_{CA}}(t_k) + \delta(t_k) \right] \tag{17}$$

The objective is to find a gain M that will modify $\Delta\delta_{cmd_{CA}}(t_k)$ in such a way that $\delta(t_{k+1}) = \delta_{cmd_{CA}}(t_k)$. Hence, it is desired to find M such that

$$\begin{aligned} \Delta\delta_{cmd_{CA}}(t_k) + \delta(t_k) = C_1\delta(t_k) + C_2\dot{\delta}(t_k) + \\ C_3 \left[M\Delta\delta_{cmd_{CA}}(t_k) + \delta(t_k) \right] \end{aligned} \tag{18}$$

Solving for M gives

$$M = \frac{\Delta\delta_{cmd_{CA}}(t_k) + (1 - C_3 - C_1)\delta(t_k) - C_2\dot{\delta}(t_k)}{C_3\Delta\delta_{cmd_{CA}}(t_k)} \tag{19}$$

Clearly, unlike the simple first-order case, this compensation is more complex. In fact, this requires not only the position of the control effector, but also the rate of change of the control effector. In practice, an estimator would be designed (for example, a Kalman filter) to estimate $\dot{\delta}(t_k)$. Referring to Figure 4, for a system with more than one actuator, **M** would be a diagonal matrix with entries along the main diagonal being $M_1, M_2, \cdots, M_m$. Here, M_i would be computed using Equation 19 and ω_{n_i}, ζ_i corresponding to the i^{th} actuator.

4. Attenuation of Zero-Order-Hold Inputs for Second-Order Actuator Dynamics With A Single Zero

The results of the previous section are now extended to the case of second-order actuator dynamics with a single zero. Let the actuator dynamics be represented by

$$\frac{\delta(s)}{\delta_{cmd}(s)} = \frac{k(s + a)}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{20}$$

In this case, it is easily seen that Equation 20 contains derivatives of the input signal. In order to eliminate $\dot{\delta}_{cmd}(t)$, introduce an intermediate variable, $z(s)$, so that

$$\frac{\delta(s)}{z(s)} \frac{z(s)}{\delta_{cmd}(s)} = \frac{k(s + a)}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{21}$$

and let

$$\frac{z(s)}{\delta_{cmd}(s)} = \frac{1}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{22}$$

and

$$\frac{\delta(s)}{z(s)} = k(s + a) \tag{23}$$

Then, it is easily seen that Equation 22, in state-space form, becomes

$$\begin{aligned} \begin{bmatrix} \dot{z}(t) \\ \ddot{z}(t) \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \begin{bmatrix} z(t) \\ \dot{z}(t) \end{bmatrix} \\ &+ \begin{bmatrix} 0 \\ 1 \end{bmatrix} \delta_{cmd}(t) \\ &= \mathbf{A} \begin{bmatrix} z(t) \\ \dot{z}(t) \end{bmatrix} + \mathbf{B}\delta_{cmd}(t) \end{aligned} \tag{24}$$

From Equation 23, the output equation becomes

$$\delta(t) = kaz(t) + k\dot{z}(t) = \begin{bmatrix} ka & k \end{bmatrix} \begin{bmatrix} z(t) \\ \dot{z}(t) \end{bmatrix} = \mathbf{C} \begin{bmatrix} z(t) \\ \dot{z}(t) \end{bmatrix} \tag{25}$$

Equations 24 and 25 describe the second-order actuator dynamics.

Using Equations 24 and 25, the discrete-time solution to the second-order actuator dynamics differential equation becomes

$$\begin{aligned} \delta(t_{k+1}) &= \left(ka\Phi_{1,1} + k\Phi_{2,1}\right) z(t_k) \\ &\quad + \left(ka\Phi_{1,2} + k\Phi_{2,2}\right) \dot{z}(t_k) + \\ &\quad \delta_{cmd}(t_k) \int_{t_k}^{t_{k+1}} ka\Phi_{1,2}(t_{k+1} - \tau)d\tau + \\ &\quad \delta_{cmd}(t_k) \int_{t_k}^{t_{k+1}} k\Phi_{2,2}(t_{k+1} - \tau)d\tau \end{aligned} \tag{26}$$

where $\mathbf{\Phi}$ was defined in Equation 14. Equation 26 can be written as:

$$\delta(t_{k+1}) = D_1 z(t_k) + D_2 \dot{z}(t_k) + \delta_{cmd}(t_k)\left[D_3 + D_4\right] \tag{27}$$

so that

$$\begin{aligned} D_1 &= ka\Phi_{1,1} + k\Phi_{2,1} \\ D_2 &= ka\Phi_{1,2} + k\Phi_{2,2} \\ D_3 &= \int_{t_k}^{t_{k+1}} ka\Phi_{1,2}(t_{k+1} - \tau)d\tau \\ D_4 &= \int_{t_k}^{t_{k+1}} k\Phi_{2,2}(t_{k+1} - \tau)d\tau \end{aligned} \tag{28}$$

It is now required to evaluate Φ and perform the math required in Equation 28. By computing the necessary inverse Laplace transforms, it is found that

$$\begin{aligned} \Phi_{1,1} &= e^{\sigma\Delta t}\left[\frac{\zeta}{\sqrt{1-\zeta^2}} sin(\omega_d\Delta t) + cos(\omega_d\Delta t)\right] \\ \Phi_{1,2} &= \frac{e^{\sigma\Delta t}}{\omega_d} sin(\omega_d\Delta t) \\ \Phi_{2,1} &= -\omega_{n^2}\frac{e^{\sigma\Delta t}}{\omega_d} sin(\omega_d\Delta t) \\ \Phi_{2,2} &= e^{\sigma\Delta t}\left[cos(\omega_d\Delta t) - \frac{\zeta}{\sqrt{1-\zeta^2}} sin(\omega_d\Delta t)\right] \end{aligned} \tag{29}$$

where ω_d and ζ were defined after Equation 16 and $\Delta t = t_{k+1} - t_k$. Using Equation 28, D_1, D_2, D_3, and D_4 are found to be

$$\begin{aligned} D_1 &= kae^{\sigma\Delta t}\left[\frac{\zeta}{\sqrt{1-\zeta^2}} sin(\omega_d\Delta t) + cos(\omega_d\Delta t)\right] \\ &\quad - k\omega_{n^2}\frac{e^{\sigma\Delta t}}{\omega_d} sin(\omega_d\Delta t) \\ D_2 &= \frac{kae^{\sigma\Delta t}}{\omega_d} sin(\omega_d\Delta t) \\ &\quad + ke^{\sigma\Delta t}\left[cos(\omega_d\Delta t) - \frac{\zeta}{\sqrt{1-\zeta^2}} sin(\omega_d\Delta t)\right] \\ D_3 &= \tfrac{ka}{\omega_d\omega_n^2}\left[\omega_d + e^{\sigma\Delta t}\left\{\sigma sin(\omega_d\Delta t) - \omega_d cos(\omega_d\Delta t)\right\}\right] \\ D_4 &= \tfrac{k}{\omega_n^2}\left[-\sigma + e^{\sigma\Delta t}\left\{\sigma cos(\omega_d\Delta t) + \omega_d sin(\omega_d\Delta t)\right\}\right] + \\ &\quad \tfrac{k\zeta}{\omega_d\omega_n}\left[-\omega_d - e^{\sigma\Delta t}\left\{\sigma sin(\omega_d\Delta t) - \omega_d cos(\omega_d\Delta t)\right\}\right] \end{aligned} \tag{30}$$

The objective is to find a gain M that will modify $\Delta\delta_{cmd_{CA}}(t_k)$ in such a way that $\delta(t_{k+1}) = \delta_{cmd_{CA}}(t_k)$. Hence, it is desired to find M such that

$$\begin{aligned} \Delta\delta_{cmd_{CA}}(t_k) + \delta(t_k) &= D_1 z(t_k) + D_2 \dot{z}(t_k) \\ &\quad + (D_3 + D_4)\left[M\Delta\delta_{cmd_{CA}}(t_k) + \delta(t_k)\right] \end{aligned} \tag{31}$$

Solving for M gives

$$M = \frac{\begin{array}{c} \Delta\delta_{cmd_{CA}}(t_k) + (1 - D_3 - D_4)\delta(t_k) \\ - D_1 z(t_k) - D_2\dot{z}(t_k) \end{array}}{(D_3 + D_4)\Delta\delta_{cmd_{CA}}(t_k)} \tag{32}$$

In this case, it can be seen that not only is the actual actuator position needed, so are the intermediate variables $z(t_k)$ and $\dot{z}(t_k)$. In practice, an estimator (for example, a Kalman filter) would be designed to provide $z(t_k)$ and $\dot{z}(t_k)$. As with the first-order and simple second-order cases, for a system with more than one actuator, **M** in Figure 4 would be a diagonal matrix with entries along the main diagonal being $M_1, M_2, \cdots, M_m$. Here, M_i would be computed using Equation 32 and ω_{n_i}, ζ_i corresponding to the i^{th} actuator.

5. Simulation Results

In this section, results from a simulation of the system displayed in Figure 4 will be shown. A rate and position constrained linear programming based control allocator will be utilized in this work. In this case, the control allocation algorithm's objective, referring to Figure 3, is to find $\boldsymbol{\delta}_{cmd}$ such that

$$\mathbf{d}_{des} = \mathbf{B}\boldsymbol{\delta}_{cmd} \tag{33}$$

where $\mathbf{B}$ is the control effectiveness matrix and $\mathbf{d}_{des}$ is typically a set of moment or acceleration commands for the roll, pitch, and yaw axes. Although, if feasible, the control allocator will be able to find a $\boldsymbol{\delta}_{cmd}$ such that Equation 33 holds, the real test is to determine what happens after the actuator dynamics operate on $\boldsymbol{\delta}_{cmd}$. Hence, the overall system goal is to achieve $\boldsymbol{\delta}$ such that

$$\mathbf{d}_{des} = \mathbf{B}\boldsymbol{\delta} \tag{34}$$

Equation 34 is the metric upon which the quality of results will be judged. In this example, four control effectors are present and the control effectiveness matrix is fixed at

$$\mathbf{B} = \begin{bmatrix} -0.4 & 0.4 & -0.1 & 0.1 \\ -0.1 & -0.1 & -0.6 & -0.6 \\ -0.1 & 0.1 & -0.1 & 0.1 \end{bmatrix} \tag{35}$$

where the elements of $\mathbf{B}$ have units of $(rad/sec^2)/deg$. Since there are more control effectors (4) than axes to control (3), a control mixed or allocator must be used. In this work, a flight-tested linear programming based control allocation algorithm with rate and position limiting will be used [2], [**?**]. It is important to note, however, that any control allocation method could be used.

In the following simulations, a mixture of actuator dynamics will be used. Specifically, the dynamics of each actuator are

$$\begin{array}{c} \frac{\delta_1(s)}{\delta_{cmd_1}(s)} = \frac{2.5(s+10)}{s^2+7.071s+25} \\ \frac{\delta_2(s)}{\delta_{cmd_2}(s)} = \frac{2.5(s+10)}{s^2+7.071s+25} \\ \frac{\delta_3(s)}{\delta_{cmd_3}(s)} = \frac{49}{s^2+7s+49} \\ \frac{\delta_4(s)}{\delta_{cmd_4}(s)} = \frac{5}{s+5} \end{array} \tag{36}$$

which gives $\zeta = \frac{\sqrt{2}}{2}$, $\omega_n = 5$, $k = 2.5$ for the second-order with zero cases, $\zeta = 0.5$, $\omega_n = 7$, $k = 49$ for the simple second-order case, and $a = 10$. Each actuator is rate and position limited by the following values

$$\begin{aligned} \boldsymbol{\delta}_{min} &= \left[\begin{array}{cccc} -1.5 & -1.5 & -1.5 & -1.5 \end{array} \right] (\text{deg}) \\ \boldsymbol{\delta}_{max} &= \left[\begin{array}{cccc} 0.4 & 1.5 & 1.5 & 1.5 \end{array} \right] (\text{deg}) \\ \dot{\boldsymbol{\delta}}_{max_{CA}} &= \left[\begin{array}{cccc} 10 & 10 & 10 & 3 \end{array} \right] \left(\frac{\text{deg}}{\text{sec}} \right) \end{aligned} \tag{37}$$

These limits were selected so that at least one position and one rate limit were in effect at some time during the simulation. This was done to show that the method developed in this work is applicable when control effectors are saturated. As will be shown, when compensation is used, actuator 1 becomes upper position limited and actuator 4 becomes rate limited as the frequency of the commands increases. The command signals, $\mathbf{d}_{des}$, consist of chirps of magnitude $0.15, 0.3,$ and 0.15 $\frac{rad}{sec^2}$ in the roll, pitch, and yaw channels and where the frequency ranged from $0.5 - 2$ Hz as a linear function of time over a 10 sec time interval.

Simulation runs were performed with and without compensation for magnitude attenuation due to actuator dynamics. Ideal conditions are when the actuator dynamics can be represented by $\frac{\delta(s)}{\delta_{cmd}(s)} = 1$. If sufficient control authority and ideal conditions exist, then the control system would achieve $\mathbf{d}_{des} = \mathbf{B}\boldsymbol{\delta}$, which is the best possible performance. Figures 5 and 6 show $\mathbf{d}_{des}$ and $\mathbf{B}\boldsymbol{\delta}$ without and with the magnitude compensation described in Equations 7, 19, and 32 applied. Clearly, when the magnitude compensation is not used (Figure 5), $\mathbf{d}_{des} \neq \mathbf{B}\boldsymbol{\delta}$, and a large error exists between these two quantities. When magnitude compensation is used (Figure 6), however, $\mathbf{d}_{des} \cong \mathbf{B}\boldsymbol{\delta}$ and near ideal performance is achieved.

Figures 7 and 8 show the control effector commands, $\boldsymbol{\delta}_{cmd_{CA}}$, and actual deflections, $\boldsymbol{\delta}$, when magnitude compensation is not utilized. As expected, the actual deflections are considerably different than the commanded deflections and this directly translates to the deviations between $\mathbf{d}_{des}$ and $\mathbf{B}\boldsymbol{\delta}$ as shown in Figure 5. On the other hand, Figures 9 and 10 display the control effector commands, $\boldsymbol{\delta}_{cmd_{CA}}$, and actual deflections, $\boldsymbol{\delta}$, when magnitude compensation is used. In this case, the actual deflections are nearly equal to the commands and as a result, $\mathbf{d}_{des} \cong \mathbf{B}\boldsymbol{\delta}$. Notice that control effector 1 is upper position limited and control effector 4 is rate limited for the last few seconds of the simulation run. Thus, using the simple gain adjustment described in Equation 19 results in the actual control deflections being equal to the commanded control deflections. It is apparent that adjusting the control effector command increments can help to mitigate adverse interactions between discrete time implementations of control allocation algorithms and actuator dynamics.

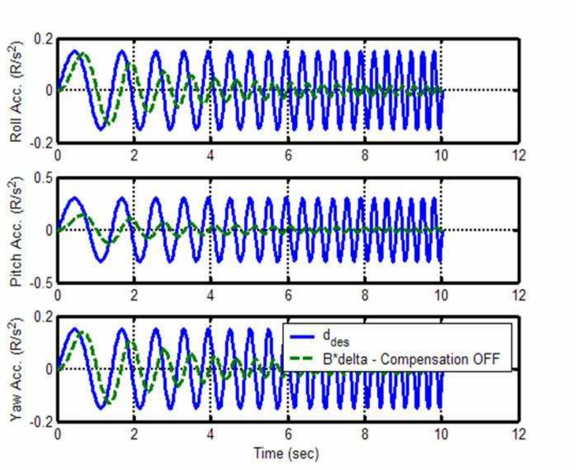

Figure 5. Commanded ($\mathbf{d}_{des}$) and simulated ($\mathbf{B}\boldsymbol{\delta}$) angular accelerations $\left(\frac{rad}{sec^2}\right)$ - Compensation OFF.

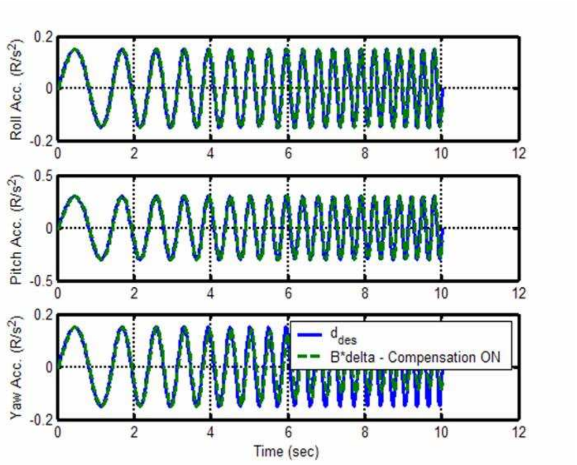

Figure 6. Commanded ($\mathbf{d}_{des}$) and simulated ($\mathbf{B}\boldsymbol{\delta}$) angular accelerations $\left(\frac{rad}{sec^2}\right)$ - Compensation ON.

6. CONCLUSIONS

Interactions between constrained control allocation algorithms and the dynamics of actuators can result in degraded performance if not carefully implemented. It was shown that the control effector commands, from a control allocation algorithm, are attenuated by actuator dynamics. One method, which can be used to extract maximum performance from such a system, is to modify the control effector command

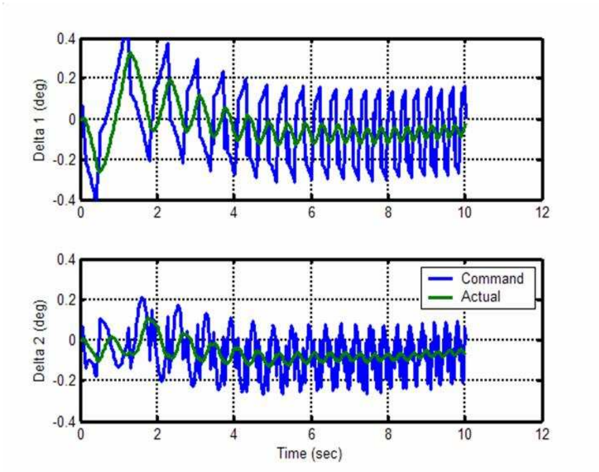

Figure 7. Control Effector 1 and 2 Positions - Compensation OFF.

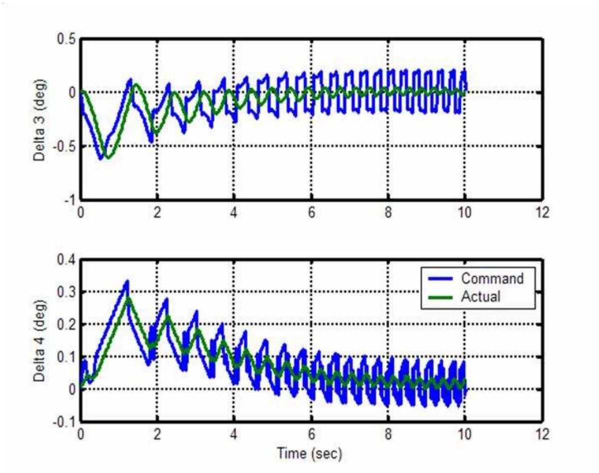

Figure 8. Control Effector 3 and 4 Positions - Compensation OFF.

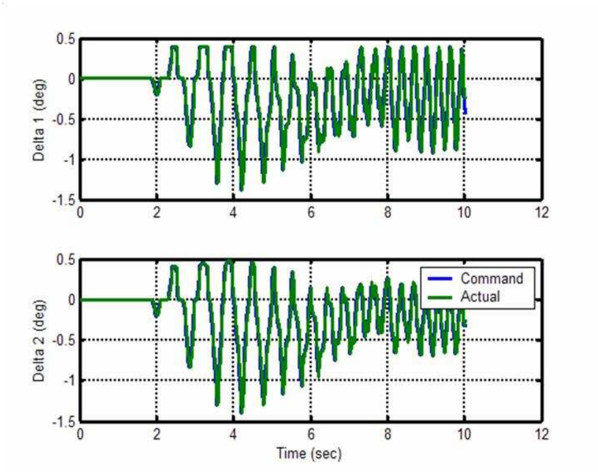

Figure 9. Control Effector 1 and 2 Positions - Compensation ON.

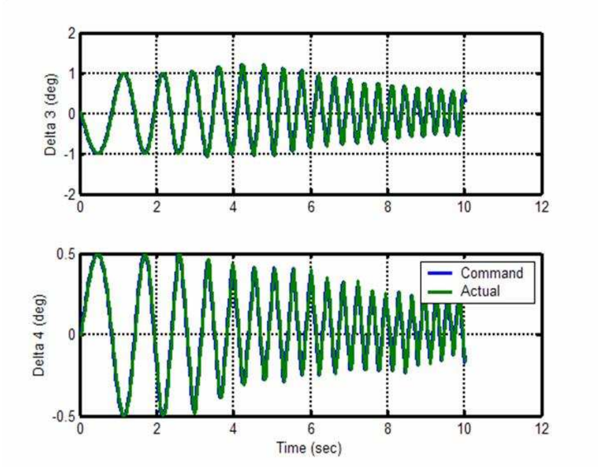

Figure 10. Control Effector 3 and 4 Positions - Compensation ON.

increments from the control allocator. The gains used to modify the commands were computed for first-order, simple second-order, and second-order with a zero actuator dynamics. Simulation results show that significant improvements can be achieved by using this method to post-process the commands from the control allocation algorithm. Benefits of this method are that the control allocation algorithm need not be modified, it requires minimal additional computations, and it is valid for saturated and unsaturated control effectors.

REFERENCES

[1] J. Buffington, “Modular Control Law Design for the Innovative Control Effectors (ICE) Tailless Fighter Aircraft Configuration 101-3,” U.S. Air Force Research Lab, Report. AFRL-VA-WP-TP-1999-3057, Wright Patterson AFB, OH, June 1999.

[2] M. Bodson, “Evaluation of Optimization Methods for Control Allocation”, *Journal of Guidance, Control and Dynamics*, volume 25, number 4, 2002, pp. 703-711.

[3] O. Härkegård, “Dynamic Control Allocation Using Constrained Quadratic Programming”, *Proceedings of the 2002 Guidance, Navigation and Control Conference*, AIAA 4761, August 2002.

[4] R. Venkataraman and M. Oppenheimer and D. Doman, “A New Control Allocation Method That Accounts for Effector Dynamics”, *Proceedings of the 2004 IEEE Aerospace Control Conference*, IEEEAC 1221, March 2004.

[5] J. Bolling, “Implementation of Constrained Control Allocation Techniques Using an Aerodynamic Model of

an F-15 Aircraft", Master's Thesis, Virginia Polytechnic Institute and State University, 1997.

[6] M. Oppenheimer and D. Doman, "Methods for Compensating for Control Allocator and Actuator Interactions", *Journal of Guidance, Control and Dynamics*, volume 27, number 5, 2004, pp. 922-927.

[7] D. Doman and M. Oppenheimer, "Improving Control Allocation Accuracy for Nonlinear Aircraft Dynamics", *Proceedings of the 2002 Guidance, Navigation and Control Conference*, AIAA 2002-4667 , August 2002.

Michael Oppenheimer *Michael W. Oppenheimer is an Electronics Engineer at the Control Design and Analysis Branch at the Air Force Research Laboratory, Wright Patterson Air Force Base, OH. He is the author or co-author of more than 25 publications including refereed conference papers, journal articles, and a technical report. He holds a Ph. D. degree in Electrical Engineering from the Air Force Institute of Technology and is a member of IEEE and AIAA. Dr. Oppenheimer's research interests are in the areas of nonlinear and adaptive control, including reconfigurable flight control and control allocation, and the application of this technology to air vehicles.*

David Doman *David B. Doman is a Senior Aerospace Engineer at the Control Design and Analysis Branch at the Air Force Research Laboratory at Wright Patterson AFB, OH. He is the Technical Area Leader for the Space Access and Hypersonic Vehicle Guidance and Control Team at AFRL. He is the author or co-author of more than 50 publications including, refereed conference papers, journal articles, technical reports and holds one US patent. He holds a Ph.D. degree in Aerospace Engineering from Virginia Tech and is currently an Associate Editor for the Journal of Guidance, Control and Dynamics, a member of the AIAA Guidance, Navigation and Control Technical Committee, an Associate Fellow of the AIAA and a Member of IEEE.*